UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/066,984	02/04/2002	Tse-Yu Yeh	5580-04403	4187

51472 7590 12/04/2006

GARLICK HARRISON & MARKISON
P.O. BOX 160727
AUSTIN, TX 78716-0727

EXAMINER	
FIEGLE, RYAN PAUL	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 12/04/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/066,984	YEH ET AL.	
	Examiner	Art Unit	
	Ryan P. Fiegle	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 October 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *21-28* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *21-28* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 21, 23-25, 27 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Avnon et al., U.S. Patent 5,559,977 in view of Computer Architecture: A Quantitative Approach by Hennessy et al. herein referred to as Hennessy.

3. As per claim 21:

Avnon teaches a processor comprising:

-An integer pipeline process integer instructions: [Integer pipeline, depicted in figure 4, IP3 and IP4. Column 7, fines 50-59]

-A floating point pipeline to process floating point instructions, wherein the floating point pipeline has a greater number of pipeline stages to process floating point instructions than a number of pipeline stages to process integer instructions in the integer pipeline: having a second number of stages, which is greater than the first number of stages, to process floating point instructions: [Floating Point pipeline, column 7, lines 35-65. Figure 4, IP1 and IP2]

-And a control circuit coupled to the integer and floating point pipelines to inhibit co-issuance of an integer instruction to the integer pipeline when the integer instruction is subsequent to two floating-point instructions in program order: [The microvector

sequencers 104u and 104v (figure 1) control the stalling of instructions issuance to both the integer and floating point pipelines. (Column 6, lines 51-65). All subsequent integer instructions are stalled and not co-issued to the execution unit, which they otherwise would be, as two integer instructions can be co-issued under normal circumstances. For the issuing inhibiting see column 7, line 50 to column 8, line 9, column 9, lines 7-39 and column 6, line 51 to column 7, line 21. For normal issuing see figure 3, column 6, lines 30-39 and column 7, lines 22-3.]

-Until the first floating point-instructions reaches a stage in the floating-point pipeline where exceptions are to be generated to ensure that the integer instruction does not graduate from the integer pipeline prior to exception determination for the floating point instructions in the floating-point pipeline, [Column 7, line 50 to column 8, line 9, column 9, lines 7-39 and column 6, line 51 to column 7, line 21. The subsequent integer instructions are not co-issued until both unsafe floating point instructions within an unsafe floating point instruction pair is deemed safe or it is handled.]

-The control circuit to also inhibit co-issuance of a second floating-point instruction that follows the first floating-point instruction in program order, if the first floating-point instruction is not a short latency floating-point instruction, to ensure that the second floating-point instruction does not graduate prior to the exception determination for the first floating point instruction, but not to inhibit the second floating point instruction from co-issuance if the first floating-point instruction is a short latency floating-point instruction, since the second floating-point instruction will not graduate prior to the exception determination for the first floating-point instruction: [Subsequent

floating point instructions are also inhibited from co-issuance when the first floating-point instruction is an "unsafe" floating-point instruction. A safe floating-point instruction is a short latency floating-point instruction since it has been determined to be not an exception causing floating-point instruction. An unsafe floating-point instruction is not a short latency floating-point instruction since it can cause an exception, which will have a longer latency to handle than the safe floating-point instructions. Column 7, fine 50 to column 8, line 9, column 9, lines 7-39 and column 6, line 51 to column 7, line 21.]

Avnon only inhibits integer instructions if both floating-point instructions in a floating-point pair are deemed unsafe. If one floating-point instruction is deemed unsafe, the integer instructions are allowed to continue. If an integer instruction reaches graduation while the one unsafe instruction produces an exception, precise exceptions will not be kept.

Hennessy states that supporting precise exceptions is a requirement in most systems. Hennessy also states that precise exceptions simplify the system interface. Further, Hennessy states that any system using demand paging or IEEE arithmetic trap handlers must use precise exceptions (Hennessy: 183).

Therefore, it would have been obvious to one of ordinary skill in the pertinent art at the time of the applicant's invention to inhibit integer instructions from issue even if one floating-point instruction in a floating-point instruction pair is deemed unsafe to ensure precise exceptions.

4. As per claim 23, Avnon teaches the apparatus of claim 21 wherein the first floating-point instruction is a long latency floating-point instruction that has a longer

latency period than the short latency floating-point instruction: [A safe floating-point instruction is a short latency floating-point instruction since it has been determined to be not an exception causing floating-point instruction. An unsafe floating-point instruction is not a short latency floating-point instruction since it can cause an exception, which will have a longer latency to handle than the safe floating-point instructions. Column 7, line 50 to column 8, line 9, column 9, lines 7-39 and column 6, line 51 to column 7, line 21.]

5. Given the similarities between claim 23 and claim 27, the arguments as stated for the rejection of claim 23 also apply to claim 27.

6. As per claim 24, Avnon teaches the apparatus of claim 21 further comprising a load/store pipeline coupled to the control circuit to process load and store instructions, the load/store pipeline also to be inhibited for co-issuance of load or store instruction, which is subsequent to the first floating-point instruction in program order, until the first floating-point instruction reaches the stage in the floating-point pipeline where exceptions are to be generated: [Column 4, lines 27-65 and column 1, line 57 to column 2, line 2. The execution units 105u and 105v and address generators 106u and 106v, together execute load/store instructions, and therefore are a load/store pipeline. Also, the load/store pipeline uses the integer pipeline, and has the same stalling technique as the integer instructions.]

7. Given the similarities between claim 24 and claim 28, the arguments as stated for the rejection of claim 24 also apply to claim 28.

8. As per claim 25, Avnon teaches a method comprising:

- Queuing a first floating-point instruction for issuance to a floating-point pipeline to process the first floating-point instruction: (Integer pipeline, depicted in figure 4, IP3 and IP4. Column 7, lines 50-59)

- Determining if exception handling for floating-point instructions is enabled: [When the FIRC 201 performs "safe instruction recognition", it determines if the exception handling is enabled, because if an instruction is determined to be "safe", exceptions are guaranteed not to happen, thus, exception handling for that instruction is disabled. If an instruction is "unsafe", exception handling is enabled. Col. 9, lines 7-15]

- Inhibiting co-issuance of an integer instruction to an integer pipeline when the integer instruction is subsequent to two floating-point instructions in program order and the exception handling is enabled, until the floating-point instructions reaches a stage in the floating-point pipeline where exceptions are to be generated to ensure that the integer instruction does not graduate from the integer pipeline prior to exception determination for the floating point instructions in the floating-point pipeline: [(Column 6, lines 51-65). All subsequent integer instructions are stalled and not co-issued to the execution unit, which they otherwise would be, as two integer instructions can be co-issued under normal circumstances. For the issuing inhibiting see column 7, line 50 to column 8, line 9, column 9, lines 7-39 and column 6, line 51 to column 7, line 21. For normal issuing see figure 3, column 6, lines 30-39 and column 7, lines 22-3.]

- Determining if the first floating-point instruction is a short latency floating-point instruction: [The FIRC 201 performs "safe instruction recognition". A safe floating-point instruction is a short latency floating-point instruction since it has been determined to be

not an exception causing floating-point instruction. An unsafe floating-point instruction is not a short latency floating-point instruction since it can cause an exception, which will have a longer latency to handle than the safe floating-point instructions. Col. 9, lines 7-15]

- Inhibiting co-issuance of a second floating-point instruction that follows the first floating-point instruction in program order and the exception handling is enabled, if the first floating-point instruction is not a short latency floating-point instruction, to ensure that the second floating-point instruction does not graduate prior to the exception determination for the first floating point instruction, but not to inhibit the second floating point instruction from co-issuance if the first floating-point instruction is a short latency floating-point instruction, since the second floating-point instruction will not graduate prior to the exception determination for the first floating-point instruction: [Subsequent floating point instructions are also inhibited from co-issuance when the first floating-point instruction is an "unsafe" floating-point instruction. A safe floating-point instruction is a short latency floating-point instruction since it has been determined to be not an exception causing floating-point instruction. An unsafe floating-point instruction is not a short latency floating-point instruction since it can cause an exception, which will have a longer latency to handle than the safe floating-point instructions. Column 7, line 50 to column 8, line 9, column 9, lines 7-39 and column 6, line 51 to column 7, line 21.]

Avnon only inhibits integer instructions if both floating-point instructions in a floating-point pair are deemed unsafe. If one floating-point instruction is deemed unsafe, the integer instructions are allowed to continue. If an integer instruction reaches

graduation while the one unsafe instruction produces an exception, precise exceptions will not be kept.

Hennessy states that supporting precise exceptions is a requirement in most systems. Hennessy also states that precise exceptions simplify the system interface. Further, Hennessy states that any system using demand paging or IEEE arithmetic trap handlers must use precise exceptions (Hennessy: 183).

Therefore, it would have been obvious to one of ordinary skill in the pertinent art at the time of the applicant's invention to inhibit integer instructions from issue even if one floating-point instruction in a floating-point instruction pair is deemed unsafe to ensure precise exceptions.

9. Claims 22 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Avnon et al., U.S. Patent 5,559,977 and Computer Architecture: A Quantitative Approach by Hennessy et al. herein referred to as Hennessy as applied to claims 21 and 25 above, and further in view of Halfhill, "SiByte Reveals 64-Bit Core for NPUs."

10. As per claim 22, Avnon and Hennessy teach the processor in claim 21, however fails to specifically teach wherein the first floating point instruction is a multiply-add instruction, that has a longer latency period than the short latency floating-point instruction.

Halfhill teaches wherein a floating-point execution unit executes floating-point multiply-add instructions. One of ordinary skill in the art would have recognized that adding the capability of a multiply-add instruction would advantageously expand the functionality of the floating-point execution unit. When the system determines that the

floating-point multiply-add instruction is unsafe, it would have a longer latency period than the short latency floating-point instruction (safe floating-point instruction).

The added functionality would have provided one of ordinary skill in the art with the motivation to add the multiply-add instruction to the possible floating-point instructions that care executable by the floating-point execution unit. This would in turn cause the second instruction to be a floating point multiply-add instruction in some instances.

11. Given the similarities between claim 22 and claim 26, the arguments as stated for the rejection of claim 22 also apply to claim 26.

Response to Arguments

12. Upon further consideration, the examiner agrees with the applicant that Avnon does not teach inhibiting the co-issuance of an integer instruction when the integer instruction is subsequent to ***a*** floating-point instruction in program order, until the first floating-point instruction reaches a stage in the floating-point pipeline where exceptions are to be generated to ensure that the integer instruction does not graduate from the integer pipeline prior to exception determination for the first floating-point instruction in the floating-point pipeline. Therefore, an additional reference has been applied to Avnon to make up for this deficiency. Since this change was not necessitated by amendment, this action is made non-final.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ryan P. Fiegle whose telephone number is 571-272-5534. The examiner can normally be reached on M-F 8-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on 571-272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Ryan P Fiegle
Examiner
Art Unit 2183